INDUCTION OF MEAN OUTPUT PREDICTION TREES FROM CONTINUOUS TEMPORAL METEOROLOGICAL DATA[1]

Dima ALBERG[2]
PhD Candidate, Department of Information Systems Engineering,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

E-mail: alberg@bgu.ac.il

Mark LAST[3]
PhD, Associate Professor, Department of Information Systems Engineering,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

E-mail: mlast@bgu.ac.il

Avner BEN-YAIR[4,5]
PhD, Department of Industrial Engineering and Management,
Sami Shamoon College of Engineering, Beer-Sheva, Israel

E-mail: avnerb@sce.ac.il

Abstract: *In this paper, we present a novel method for fast data-driven construction of regression trees from temporal datasets including continuous data streams. The proposed Mean Output Prediction Tree (MOPT) algorithm transforms continuous temporal data into two statistical moments according to a user-specified time resolution and builds a regression tree for estimating the prediction interval of the output (dependent) variable. Results on two benchmark data sets show that the MOPT algorithm produces more accurate and easily interpretable prediction models than other state-of-the-art regression tree methods.*

Key words: *temporal prediction; inductive learning; time resolution; regression trees; split criteria; multivariate statistics; multivariate time series*

1. Introduction

The time dimension is one of the most important attributes of massive continuous temporal data sets or continuous data streams [13][6], where data arrives and has to be processed on a continuous basis. Sources of such data include real-time monitoring devices as: meteorological stations, traffic control systems, financial markets, etc. If we extract a portion of data arrived over a finite time period and store it in a persistent database, it becomes a temporal dataset. Generally, the time dimension is represented in a temporal dataset as a calendar variable which has an agglomerative structure consisting of several time granules [22]: for example, 60 seconds represent 1 minute, 60 minutes represent one hour, etc. The correct choice of the pre-processing time granularity very often predetermines the accuracy and interpretability of data stream mining algorithms.

We are interested in prediction of temporal continuous variables, since they are abundant in most data streams mentioned above. However, the existing prediction methods (such as Regression Tree Models [3, 6, 18-19, 21, 24-25]) do not take the time dimension and time granularity into account, since they were developed for mainly static (time-invariant) databases. In this work, we present a new prediction algorithm capable to induce an accurate and interpretable model for estimating the values of a given continuous output variable in a massive temporal data set or a continuous data stream.

2. Problem Statement and Prior Work

In many data streams, the data is available as time-continuous statistical moments (mean, variance, etc.) calculated over pre-defined measurement cycles rather than raw values sampled at discrete points in time. Examples of such data streams include: meteorological data, financial data, factory control systems, sensor networks, etc. For example, a meteorological station might be continuously storing mean and variance estimation for a large number of meteorological attributes at predefined time intervals (e.g., every 10 minutes). A prediction model that is built using multiple statistical moments instead of discretely sampled exact values is likely to have a lower update cost, since as long as an attribute value remains within the prediction interval, fewer updates to the model will be required.

However, supervised predictive data mining models like regression models (GLM, MARS[10]) and Regression Trees (M5 [21], M5' [25], CART [3], GUIDE [19], RETIS [18], MAUVE [24], (M)SMOTI [6]) widely used at present for prediction of continuous target variables do not utilize multiple statistical moments of input and target attributes. Time-series prediction models (ARIMA, ARCH [9], and GARCH [2]), which carry out simultaneous prediction of continuous target variables represented by statistical moments are frequently non stable on significant volumes of non-stationary data and require labor-consuming reassessment at uncertain time intervals [1]. Another difficulty is to produce an interpretable set of prediction rules for such cases. Indeed, even supposing that it would be possible to build an accurate regression tree or a set of logical rules using the time dependent input attributes, the resulting model is likely to be very intricate and essentially impossible to interpret [12].

Interval prediction is an important part of the forecasting process aimed at enhancing the limited accuracy of point estimation. An interval forecast usually consists of an upper and a lower limit between which the future value is expected to lie with a prescribed probability. The limits are sometimes called forecast limits [26] or prediction bounds [5], whereas the interval is sometimes called a confidence interval [12] or a forecast region [16]. We prefer the more widely-used term prediction interval, as used by Chatfield [7] and Harvey [14], both because it is more descriptive and because the term confidence interval is usually applied to interval estimates for fixed but unknown parameters. In our case, a prediction interval is an interval estimate for an (unknown) future value of the output (dependent) variable. Since a future value can be regarded as a random variable at the time the forecast is made, a prediction interval involves a different sort of probability statement from that implied by a confidence interval. The above considerations cause a need of a model which can process the incoming data in real time and on the basis of the received results to make interval prediction of target time dependent continuous variables with a given level of statistical confidence.

2. The Proposed Data Stream Mining Methodology

2.1. The Model Induction Algorithm Overview

The proposed algorithm is aimed at inducing a prediction model for a case where every input (predictive) temporal variable is continuous and in a given sliding window k, the two statistics of mean and variance are calculated for each measurement cycle. The output (predicted) variable is the mean value of a continuous temporal variable calculated for the future sliding window $k+\Delta$.

As inputs the algorithm receives the time resolution interval j, the two first statistical moments of each temporally continuous input variable $\overline{X}$ with user predefined lag Δ history and temporally continuous output variable Y, as well as the significance level α. In the conventional regression tree algorithm, the objective is to build an inductive predictor assuming the following functional form:

$$Y = \left\{\hat{y}_{jk}\right\} = f\left\{x_{jk-\Delta}\right\} \tag{1}$$

where the predicted target variable in a sliding window k is represented as a function of input numerical variables in $k-\Delta$ sliding window, where Δ is a user-specified prediction lag parameter.

The proposed algorithm will build an inductive predictor of the following form:

$$Y = \left\{\hat{y}_{jk}, w\left(T\left(\Theta_{Y_{jk}}\right)\right)\right\} = f\left\{\overline{x}_{jk-\Delta}, \hat{s}^2_{x_{jk-\Delta}}\right\} \ , \tag{2}$$

where

$$T\left(\Theta_{Y_{jk}}\right) = w\left(\Theta_{Y_{jk}} - \overline{\Theta}_{Y_j}\right)^T \overline{\Psi}(\Theta)\left(\Theta_{Y_{jk}} - \overline{\Theta}_{Y_l}\right), \ \Theta_{Y_{jk}} \in \left\{\overline{y}_{jk}, \hat{s}^2_{y_{jk}}\right\}. \tag{3}$$

Where for time resolution j in sliding window k, $\hat{y}_{jk}$ is the predicted value of temporally continuous output variable Y, $\Theta_{Y_{jk}}$ is the mixture mean variance parameter for output variable Y, $w\left(T\left(\Theta_{Y_{jk}}\right)\right)$ is the mixture density estimation weight for the output variable Y and $\overline{x}_{jk-\Delta}$, $\hat{s}^2_{x_{jk-\Delta}}$ are mean and standard deviation estimators of a temporally continuous input variable X for time resolution j in sliding window $k-\Delta$. Finally, in (3) for time resolution j in the sliding window k, the joint variable $T\left(\Theta_{Y_{jk}}\right)$ is the mean-variance estimator based on the two first statistical moments of the output variable Y, $\overline{\Theta}_{Y_j}$ is the vector of the means of the parameter $\Theta_{Y_{jk}}$ and $\overline{\Psi}(\Theta)$ represents the within-group normalized covariance matrix for parameter $\Theta_{Y_{jk}}$, which estimates normalized mean variance covariance values of the predicted output variable Y. The confidence interval of the joint variable $T(\Theta_{Y_{jk}})$ can be approximated using the F distribution as follows:

$$UB, LB\left(T\left(\Theta_{Y_{jk}}\right)\right) = \frac{2(\ell_j - 1)(\ell_j + 1)}{\ell_j(\ell_j - 2)} \cdot F\left((\alpha/2), (1-\alpha/2), 2, \ell_j - 2\right), \tag{4}$$

where ℓ_j is the number of measurement cycles in the sliding window k for time resolution j and α is the user-specified significance level (default value is 0.05).

When the variance is independent of the mean value of the variable, the values of the joint variable $T(\Theta_{Y_{jk}})$ are expected to lie inside the confidence interval (see (4)) implying that the interaction between mean and dispersion variables does not add information about the behavior of the corresponding input variable. However, when some values of the joint variable are found outside the boundaries UB and LB (see (4)), it can be said that the interaction between mean and dispersion variables adds further information about the input variable $\overline{X}$. These outliers provide sufficient information for the output variable prediction and therefore we can consider only the outliers when evaluating the candidate split points of an input variable. In case when no outliers are found, the algorithm checks the possibility to switch to the higher time resolution and if the higher resolution represents the initial (raw) time resolution, the algorithm proceeds as the regular RETIS [18] algorithm.

The impurity merit (5.1) and (5.2) contains two parts, left and right, whereas the major objective is to find the optimal split point $\overline{X}$, which minimizes the expression in (6):

$$Var\left(T(\Theta)^L\right) = p^L \sum_{i=1}^{N_L} w_X^L \left(T(\Theta)_i^L - T\left(\overline{\Theta}\right)^L\right)^2 \tag{5.1}$$

$$\left(Var\left(T(\Theta)^R\right) = p^R \sum_{i=1}^{N_R} w_X^R \left(T(\Theta)_i^R - T\left(\overline{\Theta}\right)^R\right)^2\right) \tag{5.2}$$

$$X^* = \arg\min_{T(\Theta)} \left(Var\left(T(\Theta)^L\right) + Var\left(T(\Theta)^R\right)\right) \tag{6}$$

Here $T(\Theta)^L$ and $T(\Theta)^R$ are the left (right) joint mean variance estimator values of target variable Y, p^L and p^R are the relative number of N_L (N_R) cases that are assigned to the left (right) child, while w_X^L (w_X^R) is the left (right) mixture density estimation weight for the target variable Y.

Thus, the best split at a node is defined as the split, which minimizes the weighted variance of the joint mean variance estimator.

2. 2 Performance Metrics

A validation dataset for time resolution j and sliding window w_j is made up of $k \in \{1,...,N\}$ instances (sliding windows), each mapping an input vector $(x_1,...,x_A)^k$ to a given target y_k. The error is given by: $e_k = \hat{y}_k - y_k$, where $\hat{y}_k$ represents the predicted value for the k-th input pattern (2). The overall performance is computed by a global metric, namely the Mean Absolute Error (MAE) and Root Mean Squared (RMSE). However, the RMSE is more sensitive to high volatility errors than MAE. In order to compare the accuracy of trees from different domains the % of Explained Variability (EV) defined as:

$$EV(T) = \frac{(SSE(Mean) - SSE(T))}{SSE(Mean)} \cdot 100\% \tag{7}$$

Here $Mean$ is a majority rule predictor, which always predicts the mean value of the training set, $SSE(Mean)$and $SSE(T)$ are sum of square errors from the mean value and the value predicted by the evaluated regression tree (T) model, respectively. Another possibility to compare regression tree models is the Cost Complexity Measure (CCM) defined as:

$$CCM(T) = RMSE(T) + \alpha \cdot TS(T). \tag{8}$$

Here $RMSE(T)$ is the estimated error cost of regression tree T, $TS(T)$ is the number of terminal nodes in the tree, and α is the user defined non-negative cost complexity parameter adopted from [9], where it is shown that for a given complexity parameter, there is a unique smallest subtree of the saturated tree that minimizes the cost-complexity measure, which actually quantifies the tradeoff between the size of the tree and how well the tree fits the data.

3. Experimental Results

The performance of the MOPT algorithm proposed in the Section 2.1 was evaluated on ElNino data set from the UCI Machine Learning Repository [23]. The selected data set consist from numerical attribute types and belong to the multivariate spatio temporal regression domain. Finally, the performance of the complete Mean Output Prediction Tree (MOPT) algorithm was evaluated on the second data set, which represents a multivariate continuous data stream collected at a meteorological station in Israel during a period of about 8 years.

The algorithm performance is compared to four state-of-the-art prediction algorithms implemented by the Java API of WEKA [25]: M5P Tree [25] (Bagging M5P tree), M5-Rules [21] (Bagging M5-Rules), RepTree (Bagging RepTree) and by our implementation of the RETIS [18] algorithm (RETIS-M). The main difference between RETIS[18] and RETIS-M algorithm concludes in more fast splitting criterion implementation.

3.1. El Nino Data Set

The El Nino/Southern Oscillation (ENSO) cycle of 1982-1983, the strongest of the century, created many problems throughout the world. The El Nino dataset consists of the following attributes: buoy, date, latitude, longitude, zonal winds (west<0, east>0), meridian winds (south<0, north>0), relative humidity, air temperature and sea surface temperature. Data was taken from the buoys from as early as 1980 for some locations publicly available UCI Machine Learning Repository [23]. Other data that was taken in various locations are rainfall, solar radiation, current levels, and subsurface temperatures. The experimental data is represented using a single (daily) time resolution and it consists of 178,080 data instances. Important to note, that all data readings were taken at the same time of day and the target (predicted) variable is the subsurface temperature.

Finally in order to evaluate the predictive performance, the set of all examples was split into learning and testing examples sets in proportion 70:30.

The results in Table 1 show that under RMSE and Explained Variability criterions the MOPT and the RETIS-M algorithms are more accurate than other proposed algorithms in terms of t-test pair-wise difference. We have denoted by * the cases where the p value of the difference between MOPT and other algorithms is smaller than or equal to 5%. The MOPT algorithm outperforms significantly the other algorithms in the terms of cost complexity measure. Finally, we will to consider that our proposed MOPT Tree is more interpretable than RETIS-M tree in terms of Tree Size measure (7 vs. 23).

Table 1. El Nino data set learners comparison

Learner	RMSE	TS	CCM	EV
B-M5 Rules	0.84*	7	1.01*	0.46*
B-M5P Tree	0.83*	10	1.07*	0.47*
B-REPTree	1.57*	5	1.69*	NA
M5 Rules	0.86*	7	1.03*	0.45*
M5P Tree	0.84*	8	1.03*	0.46*
MOPT	**0.60**	7	0.77	0.62
REPTree	1.57*	3	1.64*	NA
RETIS-M	**0.63**	23	1.18*	**0.60**

3.2. Israel Meteorology Data Set

In this experiment we used the data collected at a meteorological station in Israel during a period of about 8 years (from 01/07/1997 to 31/08/2005). Spatio- temporal meteorological attributes (such as pressure, temperature, solar radiation, horizontal wind: direction, speed, gust speed, gust time, and vertical wind: down-up and up direction) are measured constantly in time and saved every 10 minutes in the form of mean and variance. The selected data set exceeds 1,500,000 records. The total number of temporal and meteorological attributes collected at the three stations is 22. Our first experiment was run on the summer months (JUN, JUL and AUG) only. The experimental data was represented using 5 time resolutions (10, 30, 60, 90 and 120 Minutes). The algorithms were run for 11:00-12:00 and 23:00 – 24:00 hours prediction.

The aim of this experiment was to compare the different state-of-the-art algorithms for different time resolutions in order to be able to predict wind directions for short time range (now-casting) up to 8 hours sliding window ahead. We have shown that the most state-of-the-art algorithms gave the same or poorer quality of results and less interpretable trees as the proposed Mean Output Prediction Tree (MOPT) algorithm. The results also pinpoint the fact that sometimes there was no need to use very high time resolution, but only lower time resolutions, since the statistical measures checked (i.e. RMSE, Cost Complexity Measure and Percentage of Explained Variability) were similar.

Tables 2, 3 compare between the proposed MOPT algorithm and four state-of-the-art algorithms: Modified RETIS (RETIS-M), M5P and REPTree in five time resolution scales in terms of Cost Complexity and Explained Variability measures. For more effective evaluation of the MOPT algorithm we performed short term prediction of 11:00 and 23:00 hour. The sliding window size and the prediction lag were set to 8 hours and 3 hours, respectively. Thus for predicting 11 hour wind direction we collected data from 00:00 to 8:00 and for predicting 23 hour wind direction we collected data from 12:00 to 20:00. In each prediction case, the main issue is to predict wind direction 3 hours ahead therefore fast robust and accurate prediction algorithm producing a compact model is needed. For each time resolution, we preprocessed the raw data and calculated the first two statistical moments for each attribute in every measurement cycle. The MOPT algorithm refers to each input attribute as a 2-dimensional array (two moments × number of instances) and determines the split point with the aid of two moments target variable impurity and the variance of the input variable. As in the previous experiments, the differences are considered statistically significant when the p-value of the t-pair-wise test statistic is smaller than or equal to 5% which signed by *.

Table 2. MOPT and state-of-the-arts models cost complexity Measure (CCM) cross resolutions results for 11:00 hour prediction

TR	MOPT	RETIS-M	M5P	REPTree
10	116.61	227.28*	117.18	124.23
30	117.20	245.48*	149.48*	132.07
60	120.58	251.13*	172.78*	143.31
90	120.32	236.58*	171.03*	139.11
120	114.61	240.73*	188.88*	148.70*

In 10 minutes resolution the M5P slightly outperforms the proposed MOPT model and significantly better than other models. In other resolutions the MOPT model significantly better than other state-of-arts models. This result pinpoint to the fact that adding second moments to split criterion improves quality of prediction for higher time resolution.

Table 3. MOPT and state-of-the-arts models cost complexity measure (CCM) cross resolutions results for 23:00 hour prediction

TR	MOPT	RETIS-M	M5P	REPTree
10	43.95	60.04*	55.00	53.01
30	43.27	59.84*	60.53*	59.74*
60	45.45	62.28*	53.38	60.72*
90	44.55	59.76*	57.76	74.45*
120	44.53	61.82*	53.60	60.00*

By comparison to state-of-the-art algorithms, the MOPT algorithm demonstrates more stable prediction accuracy with a more compact tree size in 23:00 hour prediction (for example in 10 minutes resolution the size of MOPT tree is 502 versus 2947 of M5P). In this case, the regression tree pruning procedure may significantly reduce the final size of the tree, but this procedure is out of scope in the proposed MOPT approach because our main purpose is to build accurate and compact tree with minimal access to the sliding window training data.

Table 4. MOPT and state-of-the-arts models explained variability (%EV) cross resolutions results for 11:00 hour prediction

TR	MOPT	RETIS-M	M5P	REPTree
10	45.24%	32.46%	60.18%	59.96%
30	45.35%	25.35%	41.99%	50.26%
60	44.59%	25.78%	29.73%	43.43%
90	44.32%	30.45%	29.69%	53.98%
120	48.65%	32.39%	23.09%	37.34%

Table 5. MOPT and state-of-the-arts models explained variability (%EV) cross resolutions results for 23:00 hour prediction

TR	MOPT	RETIS-M	M5P	REPTree
10	29.3%	-6.1%	22.7%	8.7%
30	30.7%	-5.6%	-10.4%	12.1%
60	25.6%	-14.7%	8.5%	12.7%
90	28.5%	-4.8%	-13.3%	-43.9%
120	28.3%	-13.3%	12.9%	16.4%

The final stage of this experiment presented in the Tables 4-5 demonstrates the comparison of Percentage of Explained Variability EV between five defined models and time resolutions for 11:00 and 23:00 Hours respectively. The cells with negative explained variability percent indicate the fact that the induced model is poorer (less accurate) than a simple majority rule mean model. For example, RETIS-M, M5P and REPTree models have not contributed to the explained variability of the 23:00 hour prediction. Important to emphasize, that three state-of-the-art algorithms did not scale well to the low time resolution of 120 minutes.

4. Conclusions

In this work, we have presented the two moments (mean-variance) Mean Output Prediction Tree algorithm (MOPT), which is able to predict large amounts of massive temporal data sets. The proposed algorithm differs from the state-of-the-art regression algorithms in the splitting of each input and output feature to two moments according to the input time resolution and it can also identify the most appropriate prediction time resolution that minimizes the prediction error and builds more compact interval based regression tree.

The two conducted experiments indicate that the proposed algorithm produces more accurate and compact models by comparison to the modern state-of-the-art regression tree algorithms.

5. References

1. Alberg, D., Shalit, H. and Yosef, R. **Estimating stock market volatility using asymmetric GARCH models**, Applied Financial Economics, vol. 18(15), 2008, pp. 1201-1208
2. Bollerslev, T.A. **Conditionally heteroskedastic time series model for speculative prices and rates of return**, Review of Economics and Statistics, vol. 69, 1987, pp. 542–547
3. Breiman, L. and Friedman, J. **Estimating Optimal Transformations for Multiple Regression and Correlation**, J. Amer. Statist. Assoc., vol. 80, 1985, p. 580
4. Breiman, L., Friedman, J., Olshen, R. and Stone, C. **Classification and Regression Trees**, Wadsworth Int. Group, Belmont California USA, 1984
5. Brockwell, P. and Davis, R. **Time Series: Theory and Methods**, 2nd ed., New York: Springer-Verlag, 1991
6. Ceci, M., Appice, A. and Malerba, D. **Comparing simplification methods for model trees with regression and splitting nodes**, Foundations of Intelligent Systems, 14th International Symposium, vol. 2871 of LNAI, 2003, pp. 49–56
7. Chatfield, C. **The Analysis of Time Series**, 5th ed., London: Chapman and Hall, 1995, pp. 431-441
8. Cortez, F. and Morais, R.D. **A data mining approach to predict forest fires using meteorological data**, in Neves, J., Santos, M.F. and Machado, J. (eds.) "New Trends in Artificial Intelligence", Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, 2007, pp. 512-523
9. Engle, R. **Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation**, Econometrica, vol. 50, 1982, pp. 987-1007
10. Friedman, J. **Multivariate adaptative regression splines**, Annals of Statistics, 1991, pp. 1-19
11. Gama, J., Rocha, R., Medas, P., Wah, B. and Wang, J. **Accurate decision trees for mining high-speed data streams**, KDD 2003, 2003 , pp. 523-528
12. Granger, C. and Newbold, P. **Forecasting Economic Time Series**, 2nd ed., Academic Press, New-York, 1986
13. Han, J., Cai, D. and Chen, Y. **Multi dimensional analysis of data streams using stream cubes in data streams models and algorithms**, ed. Aggarwal, C., 2007, pp. 103-125
14. Harvey, A. **Forecasting Structural Time Series Models and the Kalman Filter**, C U P, Cambridge, 1989
15. Hulten, G., Spencer, L. and Domingos, P. **Mining time-changing data stream**, Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 97-106
16. Hyndman, R. **Highest-density forecast regions for non-linear and non-normal time series models**, Journal of Forecasting, vol. 14, 1996
17. Kamber, M. and Han, J. **Data Mining: Concepts And Techniques**, 2nd ed., The Morgan Kaufmann Series in Data Management Systems, 2007
18. Karalic, A. **Linear regression in regression tree leaves**, in "Proceedings of International School for Synthesis of Expert", vol. 10(3), 1992, pp. 151-163
19. Loh, W.Y. **Regression tree models for designed experiments**, 2nd Lehmann Symposium, Knowledge, 2006, pp. 210-228
20. Ma, J., Zeng, D. and Chen, H. **Spatial-temporal cross-correlation analysis: a new measure and a case study in infectious disease**, in Mehrotra, S. *et. al.* (eds) Informatics, ISI, LNCS 3975, 2006, pp. 542-547
21. Quinlan, J. **Learning with continuous classes**, in "Proceedings of the 5th Australian Joint Conference on Artificial Intelligence", Singapore World Scientific, 1992
22. Spranger, S. and Bry, F. **Temporal data modeling and reasoning for information statistical view of boosting**, The Annals of Statistic, vol. 38(2), 2006, pp. 337-374

23. Vens, C. and Blockeel, H. **A simple regression based heuristic for learning model trees**, Intelligent Data Analysis, vol. 10(3), 2006, pp. 215-236

24. Wang, Y. and Witten, I. **Inducing of model trees for predicting continuous classes**, in "Proceedings of the 9th European Conference on Machine Learning", Springer-Verlag, 1997, pp. 128-137

25. Wei, S., Dong, Y. and Pei, J.-B. **Time Series Analysis**, Redwood City Cal Addison-Wesley, 1990

26. * * * **UCI, Machine Learning Repository**, http://archive/ics/uci edu/ml/index html

[1] **Acknowledgment:** This study was partially supported under a research contract from the Israel Ministry of Defense.

[2] **Dima Alberg** is currently Ph.D. candidate in the Department of Information Systems Engineering, Ben-Gurion University of the Negev under the supervision of Prof. Mark Last. He is also Engineer in the Industrial Engineering and Management Department of SCE - Shamoon College of Engineering. Dima was born in Vilnius, and repatriated to Israel in 1996. He received his B.A and M.A. in Economics and Computer Science from Ben-Gurion University of the Negev. His current research interests are in time series data mining, data streams segmentation and computer simulation. His recent publications include the Journal of Business Economics and Management, Applied Financial Economics, and Communications in Dependability and Quality Management.

[3] Mark Last is currently Associate Professor at the Department of Information Systems Engineering, Ben-Gurion University of the Negev, Israel and Head of the Software Engineering Program. Prior to that, he was a Visiting Research Scholar at the US National Institute for Applied Computational Intelligence, Visiting Assistant Professor at the Department of Computer Science and Engineering, University of South Florida, USA, Senior Consultant in Industrial Engineering and Computing, and Head of the Production Control Department at AVX Israel. Mark obtained his Ph.D. degree from Tel-Aviv University, Israel in 2000. He has published over 140 papers and chapters in scientific journals, books, and refereed conferences. He is a co-author of two monographs and a co-editor of seven edited volumes. His main research interests are focused on data mining, cross-lingual text mining, software testing, and security informatics.

Prof. Last is a Senior Member of the IEEE Computer Society and a Professional Member of the Association for Computing Machinery (ACM). He currently serves as Associate Editor of IEEE Transactions on Systems, Man, and Cybernetics, where he has received the Best Associate Editor Award for 2006, and Pattern Analysis and Applications (PAA). Prof. Last is very active in organizing cooperative international scientific activities. He has co-chaired four international conferences and workshops on data mining and web intelligence.

[4] **Avner Ben-Yair** is lecturer in the Industrial Engineering and Management Department, Sami Shamoon College of Engineering, Israel. He was born in Moscow in 1961. He received his B.Sc. in Mechanical Engineering from the Moscow Polygraphic Institute, Russia, and his M.Sc. degree in Health and Safety Engineering and Management (Summa Cum Laude) from the Ben Gurion University of the Negev, Israel. He also received his Ph.D. degree in Industrial Engineering and Management from the Ben Gurion University of the Negev, Israel. His professional experience includes 13 years of engineering and management positions in Israeli chemical, pharmaceutical and high-tech industries. His current research interests are in economic aspects of safety, reliability and failure analysis, trade-off optimization models for organization systems, production planning, scheduling and control, cost optimization and PERT-COST models, and strategic management. He has published 40 articles in various scientific sources. His recent publications have appeared in Mathematics and Computers in Simulation, International Journal of Production Economics, Communications in Dependability and Quality Management, and Computer Modelling and New Technologies.

[5] Corresponding author

[6] Codification of references:

[1]	Alberg, D., Shalit, H. and Yosef, R. **Estimating stock market volatility using asymmetric GARCH models**, Applied Financial Economics, vol. 18(15), 2008, pp. 1201-1208
[2]	Bollerslev, T.A. **Conditionally heteroskedastic time series model for speculative prices and rates of return**, Review of Economics and Statistics, vol. 69, 1987, pp. 542–547
[3]	Breiman, L. and Friedman, J. **Estimating Optimal Transformations for Multiple Regression and Correlation**, J. Amer. Statist. Assoc., vol. 80, 1985, p. 580
[4]	Breiman, L., Friedman, J., Olshen, R. and Stone, C. **Classification and Regression Trees**, Wadsworth Int. Group, Belmont California USA, 1984
[5]	Brockwell, P. and Davis, R. **Time Series: Theory and Methods**, 2nd ed., New York: Springer-Verlag, 1991
[6]	Ceci, M., Appice, A. and Malerba, D. **Comparing simplification methods for model trees with regression and splitting nodes**, Foundations of Intelligent Systems, 14th International Symposium, vol. 2871 of LNAI, 2003, pp. 49–56